PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application No.: 09/675,380
Confirmation No.: 1411
Filing Date: September 29, 2000

Inventors: Weber et al.

Title: BUSINESS PROCESS
FRAMEWORK FOR
REINSURANCE

Examiner: Bleck, C.
Art Unit: 3626
Atty. Dkt. No: 5053-28301

CERTIFICATE OF ELECTRONIC TRANSMISSION
UNDER 37 C.F.R. §1.8

DATE OF DEPOSIT: 2/6/08

I hereby certify that this correspondence is being deposited with the United States Patent Office electronic filing system on the date indicated above

Jackie Pitre
Jackie L. Pitre

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Appellant submits the following Appeal Brief in support of claims 68, 69, 71-77, 79-87, 90-92, 95, 96, 98-104, 106-114, and 117-119 of the above-referenced application. Appellant submits that each of these claims is patentable and in condition for allowance.

I. Real Party in Interest

The Real Party in Interest for the appealed application is Computer Sciences Corporation, a corporation having a place of business at 200 West Cesar Chavez, Austin, Texas 78701.

II. Related Appeals and Interferences

There are no related appeals or interferences that will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

III. Status of Claims

Claims 1-119 have been entered in the case. Claims 1-67, 70, 78, 88, 89, 93, 94, 97, 105, 115, and 116 have been cancelled. Claims 68, 69, 71-77, 79-87, 90-92, 95, 96, 98-104, 106-114, and 117-119 are pending. Claims 68, 69, 71-77, 79-87, 90-92, 95, 96, 98-104, 106-114, and 117-119 have been rejected. No claims have been allowed. Claims 68, 69, 71-77, 79-87, 90-92, 95, 96, 98-104, 106-114, and 117-119 are being appealed.

IV. Status of Amendments

An Office Action was mailed on August 30, 2007. No amendments have been made to the claims since the mailing of this Office Action.

V. Summary of Claimed Subject Matter

Reinsurance includes the transfer by a first insurer of all or part of a risk to a second insurer to provide protection against the risk. The complexity of the reinsurance field tends to require software for reinsurance administration to be complex as well. Software for reinsurance administration may be expected to handle risk selection, portfolio analysis, policy administration, claims, accounting, and other areas vital to the reinsurance field. Reinsurance profits may depend on analysis of historical information, the ability to predict trends, and the ability to identify cumulative exposures within a current portfolio, and reinsurance software may therefore be expected to meet requirements relating to those functions. Consequently, the development of a software system for reinsurance administration to meet the above-identified needs may require great time and expense. (See page 1, line 23 to page 2, line 15).

Appellant developed an improved method and system for developing and a reinsurance administration system using software frameworks.

Independent claim 68 is directed to a computer readable medium that includes program instructions that are computer-executable to implement a method for developing a reinsurance administration system for reinsurance contracts. (See page 1, lines 18-23). The method includes obtaining a reinsurance business process framework including common functionality for one or more reinsurance business processes. (See page 4, lines 6-8). The reinsurance business process framework includes one or more classes of objects designed for a reinsurance administration system; a plurality of support processes, a plurality of hook methods, and a designated order for executing steps in one or more application programs. (See page 37, line 24; page 30, lines 4-29; page 31, line 16 to page 32, lines 21-25). The steps include pre-execution, data entry, data validation, pre-commission, commission, and post-commission. (See page 32, lines 21-25). One or more reinsurance business process subclasses are created from the classes of objects of the reinsurance business process framework. (See page 31, lines 16-22). The reinsurance business process subclasses inherit one or more of the hook methods of the reinsurance business process framework. One or more of the support processes are associated with one or more of the reinsurance business process subclasses. (See page 30, lines 4-16). At least one of the hook methods of the reinsurance business process framework is overridden to access a stage in an execution of one of the reinsurance business processes and to identify a support process to be executed. Overriding the hook method includes overriding a method to be executed during data entry. (See page 32, lines 1-19). One or more reinsurance business process subclasses are combined to build one or more application programs for the reinsurance administration system. (See page 33, lines 4-8). The order for executing steps in the one or more application programs is the order for the reinsurance business process framework. (See page 31, lines 21-25).

The method further includes creating one or more reinsurance contract objects are created that represent one or more reinsurance contracts. (See page 37, lines 23-29). Creating a reinsurance contract object includes: identifying one or more inheritable contract objects from the class of objects to represent one or more conditions of a reinsurance contract, creating an instance of the inheritable contract object to identify a condition object, and configuring properties and methods of the condition object consistent with the reinsurance contract. (See page 37, line 4 to page 38, line 8). The reinsurance contract includes the transfer by a first insurer of at least a portion of the risk associated with a primary insurance contract to a second insurer to provide protection to the first insurer against the risk associated with the primary insurance contract. (See page 1, lines 19-21). The reinsurance contract object is a parent of a section object. The condition object is a child of the section object. (See page 37, line 21 – page 38, line 8). Reinsurance process objects as defined by the combined reinsurance business process subclasses are automatically generated when one or more of the application programs are initiated. (See page 4, lines 13-18, page 33, lines 6-8). At least of the application programs for the reinsurance administration system is executed. (See page 9, lines 11-14).

Independent claim 95 is directed to a method that includes obtaining a reinsurance business process framework including common functionality for one or more reinsurance business processes. (See page 4, lines 6-8). The reinsurance business process framework includes one or more classes of objects designed for a reinsurance administration system; a plurality of support processes, a plurality of hook methods, and a designated order for executing steps in one or more application programs. (See page 37, line 24; page 30, lines 4-29; page 31, line 16 to page 32, line 21-25). The steps include pre-execution, data entry, data validation, pre-commission, commission, and post-commission. (See page 32, lines 21-25). One or more reinsurance business process subclasses are created from the classes of objects of the reinsurance business process framework. (See page 31, lines 16-22). The reinsurance business process subclasses inherit one or more of the hook methods of the reinsurance business process framework. One or more of the support processes are associated with one or more of the reinsurance business process subclasses. (See page 30, lines 4-16). At least one of the hook methods of the reinsurance business process framework is overridden to access a stage in an execution of one of the reinsurance business processes and to identify a support process to be executed. Overriding the hook method includes overriding a method to be executed during data entry. (See page 32, lines 1-19). One or more reinsurance business process subclasses are combined to build one or more application programs for the reinsurance administration system. (See page 33, lines 4-8). The order for executing steps in the one or more application programs is the order for the reinsurance business process framework. (See page 31, lines 21-25).

The method further includes creating one or more reinsurance contract objects are created that represent one or more reinsurance contracts. (See page 37, lines 23-29). Creating a reinsurance contract object includes: identifying one or more inheritable contract objects from the class of objects to represent one or more conditions of a reinsurance contract, creating an instance of the inheritable contract object to identify a condition object, and configuring properties and methods of the condition object consistent with the reinsurance contract. (See page 37, line 4 to page 38, line 8). The reinsurance contract includes the transfer by a first insurer of at least a portion of the risk associated with a primary insurance contract to a second insurer to provide protection to the first insurer against the risk associated with the primary insurance contract. (See page 1, lines 19-21). The reinsurance contract object is a parent of a section object. The condition object is a child of the section object. (See page 37, line 21 – page 38, line 8). Reinsurance process objects as defined by the combined reinsurance business process subclasses are automatically generated when one or more of the application programs are initiated. (See page 4, lines 13-18, page 33, lines 6-8). At least of the application programs for the reinsurance administration system is executed. (See page 9, lines 11-14).

VI. Grounds of Rejection to be Reviewed on Appeal

Claims 68, 69, 71-77, 79-96, 98-104, and 106-119 are rejected under 35 U.S.C. 103(a) as obvious over U.S. Patent 6,694,506 to LeBlanc et al. ("Leblanc") in view of U.S. Patent No. 5,946,694 to Copeland et al. ("Copeland"), Pree (Wolfgang Pree, Meta Patterns – A means for capturing the essential of reusable object-oriented design, Proceedings, ECOOP'94, 1994 – info.unikarlsruhe.de, and U.S. Patent No. 6,049,773 to McCormack et al. ("McCormack").

VII. Argument

Claims 68, 69, 71-77, 79-96, 98-104, and 106-119 were rejected under 35 U.S.C. 103(a) as obvious over Leblanc in view of Copeland, Pree, and McCormack. Appellant traverses this rejection for the following reasons.

Claim 68

In order to reject a claim as obvious, the Examiner has the burden of establishing a *prima facie* case of obviousness. *In re Warner* et al., 379 F.2d 1011, 154 U.S.P.Q. 173, 177-178 (C.C.P.A. 1967). All words in a claim must be considered in judging the patentability of that claim against the prior art. *In re Wilson*, 424 F.2d 1382, 1385 (CCPA 1970).

Claim 68 is directed to a method that includes:

> obtaining a reinsurance business process framework, wherein the reinsurance business process framework comprises common functionality for one or more reinsurance business processes and wherein the reinsurance business process framework comprises: one or more classes of objects designed for a reinsurance administration system; a plurality of support processes, a plurality of hook methods, and a designated order for executing steps in one or more application programs, wherein the steps comprise pre-execution, data entry, data validation, pre-commission, commission, and post-commission

Claim 68 further describes:

> combining one or more reinsurance business process subclasses to build one or more application programs for the reinsurance administration system, wherein the order for executing steps in the one or more application programs is the order for the reinsurance business process framework

Claim 68 further describes:

> executing at least of the one or more application programs for the reinsurance administration system

Claim 68 is directed to a computer-readable medium including program instructions for developing a reinsurance administration system for reinsurance contracts. The program instructions are executable to implement a method that includes obtaining a reinsurance framework. The reinsurance framework includes classes of objects designed for a reinsurance administration system and a designated order for executing steps in one or more applications programs, the steps including pre-execution, data entry, data validation, pre-commission, commission, and post-commission. Reinsurance business process subclasses are created from the classes of objects and combined to build one or more application programs for the reinsurance administration system. The order for executing steps in the application programs is the designated order for the framework. The method further includes executing one or more of the application programs.

The Examiner does not appear to provide any prior art reference disclosing a framework that includes a designated order for executing steps in an application program, the steps including pre-execution, data entry, data validation, pre-commission, commission, and post-commission. The Examiner states:

> this method would be performed the same regardless of whether the method had a specific type of framework support process, support process, or reinsurance framework. Thus, this descriptive material will not distinguish the claimed invention from the prior art in terms of patentability, see In re Gulack, 703 F.2cd 1381, 1385, 217 USPQ 1401 404 (Fed. Cir. 1983); In re Lowry, 32 F.3d 1579, 1583-84, 32 USPQ2d 1031, 1035 (Fed. Cir. 1994). For further guidance, not MPEP §2106, common situations involving nonfunctional descriptive material are: "a process that differs from the prior at only with respect to nonfunctional descriptive material that cannot alter how the process steps are to be performed to achieve the utility of the invention."

In the *Lowery* case, the Federal Circuit held that the Board erred by denying weight to patentable weight to data structure limitations. The court stated:

> As part of its burden to establish a *prima facie* case of obviousness, *see In re Oetiker,* 977 F.2d 1443, 1445, 24 USPQ2d 1443, 1444 (Fed.Cir.1992), the burden of establishing the absence of a novel, nonobvious functional relationship rests with the PTO. "If examination at the initial stage does not produce a *prima facie* case of unpatentability, then without more the applicant is entitled to grant of the patent." *Id.*

With respect to claim 68, the designated order of steps, including the steps of pre-execution, data entry, data validation, pre-commission, commission, and post-commission, are not merely descriptive material because they determine how the steps of the method are carried out when the application programs are executed. As noted above, claim 68 recites in part: "combining one or more reinsurance business process subclasses to build one or more application programs for the reinsurance administration system, wherein the order for executing steps in the one or more application programs is the order for the reinsurance business process framework" and "executing at least one of the one or more application programs for the reinsurance administration system." Appellant submits that the designated order for executing steps in one or more application programs, the steps including pre-execution, data entry, data validation, pre-commission, commission, and post-commission, as recited in claim 68, determines how the steps are carried out when the application program(s) are executed as part of the method. Thus the order for executing steps in one or more application programs, including the steps of pre-execution, data entry, data validation, pre-commission, commission, and post-commission, are not non-functional descriptive material. Appellant submits that the Examiner has not made a *prima facie* case of unpatentability with respect to claim 68.

For at least the reasons stated above, Appellant submits that claim 68 is allowable over the cited art.

Claim 95

Claim 95 includes, but is not limited to, the feature of:

> obtaining a reinsurance business process framework, wherein the reinsurance business process framework comprises common functionality for one or more reinsurance business processes and wherein the reinsurance business process framework comprises: one or more classes of objects designed for a reinsurance administration system; a plurality of support processes; a plurality of hook methods, and a designated order for executing steps in one or more application programs, wherein the steps comprise pre-execution, data entry, data validation, pre-commission, commission, and post-commission
>
> . . .
>
> combining one or more reinsurance business process subclasses to build one or more application programs for the reinsurance administration system, wherein the order for executing steps in the one or more application programs is the order for the reinsurance business process framework
>
> . . .
>
> executing at least of the one or more application programs for the reinsurance administration system

For at least the same reasons cited above for claim 68, Appellant submits that claim 95 is patentable over the cited art.

Claim 73

Claim 73 recites, in part, "wherein the at least one hook method that is overridden comprises a method to be executed prior to execution of a data entry support process." Appellant submits that the cited art does not appear to teach or suggest this feature, in combination with the features of independent claim 68, for at least the reasons cited above. Further, with respect to claim 73, the Examiner states:

> It is respectfully submitted that while LeBlanc, Copeland, and Pree do not disclose overriding every hook method as recited in claims 72-77, Copeland does disclose that they can be used before an object performs its method and Pree discloses that a hook method can be used at any location in a routine or program and that they can be overridden. The motivation being for the purpose of debugging or enhancing functionality.

The Examiner appears to acknowledge that the cited art does not disclose the above quoted features of claim 73. The Examiner appears to rely merely on a blanket motivation for "debugging or enhancing functionality." Appellant submits, however, that the purpose of debugging or enhancing functionality does not provide a motivation to modify LeBlanc, Copeland, and Pree to override a hook method to be executed prior to execution of a data entry support process.

Claim 74

Claim 74 recites, in part, "wherein the at least one hook method that is overridden comprises a method to be executed during execution of data entry without a user interface." Appellant submits that the cited art does not appear to teach or suggest this feature, in combination with the features of independent claim 68, for at least the reasons cited above. In particular, Appellant submits that the purpose of "debugging or enhancing functionality" does not provide a motivation to modify LeBlanc, Copeland, and Pree to override a hook method to be executed during execution of data entry without a user interface.

Claim 75

Claim 75 recites, in part, "wherein the at least one hook method that is overridden comprises a method to be executed prior to execution of a database commit support process." Appellant submits that the cited art does not appear to teach or suggest this feature, in combination with the features of independent claim 68, for at least the reasons cited above. In particular, Appellant submits that the purpose of "debugging or enhancing functionality" does not provide a motivation to modify LeBlanc, Copeland, and Pree to override a hook method to be executed prior to execution of a database commit support process.

Claim 76

Claim 76 recites, in part, "wherein the at least one hook method that is overridden comprises a method to be executed prior to execution of a database rollback support process." Appellant submits that the cited art does not appear to teach or suggest this feature, in combination with the features of independent claim 68, for at least the reasons cited above. In particular, Appellant submits that the purpose of "debugging or enhancing functionality" does not provide a motivation to modify LeBlanc, Copeland, and Pree to override a hook method to be executed prior to execution of a database rollback support process.

Claim 100

Claim 100 recites, in part, "wherein the at least one hook method that is overridden comprises a method to be executed prior to execution of a data entry support process." Appellant submits that the cited art does not appear to teach or suggest this feature, in combination with the features of independent claim 95, for at least the reasons cited above with respect to claim 95 and claim 73.

Claim 101

Claim 101 recites, in part, "wherein the at least one hook method that is overridden comprises a method to be executed during execution of data entry without a user interface." Appellant submits that the cited art does not appear to teach or suggest this feature, in combination with the features of independent claim 95, for at least the reasons cited above with respect to claim 95 and claim 74.

Claim 102

Claim 102 recites, in part, "wherein the at least one hook method that is overridden comprises a method to be executed prior to execution of a database commit support process." Appellant submits that the cited art does not appear to teach or suggest this feature, in combination with the features of independent claim 95, for at least the reasons cited above with respect to claim 95 and claim 75.

Claim 103

Claim 103 recites, in part, "wherein the at least one hook method that is overridden comprises a method to be executed prior to execution of a database rollback support process." Appellant submits that the cited art does not appear to teach or suggest this feature, in combination with the features of independent claim 95, for at least the reasons cited above with respect to claim 76.

VIII. Conclusion

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 713-726 was erroneous, and reversal of his decision is respectfully requested.

If any extension of time is required, Appellant hereby requests the appropriate extension of time. If any fees are omitted or if any additional fees are required or have been overpaid, please appropriately charge or credit those fees to Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C. Deposit Account Number 50-1505/5053-28301/EBM.

Respectfully submitted,

Eric B. Meyertons
Reg. No. 34,876
Attorney for Appellant

MEYERTONS, HOOD, KIVLIN, KOWERT & GOETZEL, P.C.
P.O. Box 398
Austin, TX 78767-0398
(512) 853-8800 (voice)
(512) 853-8801 (facsimile)

Date: 2/6/08

IX. Claims Appendix

The claims on appeal are as follows:

68. A computer readable medium comprising program instructions for developing a reinsurance administration system for reinsurance contracts, wherein the program instructions are computer-executable to implement a method of:

obtaining a reinsurance business process framework, wherein the reinsurance business process framework comprises common functionality for one or more reinsurance business processes and wherein the reinsurance business process framework comprises: one or more classes of objects designed for a reinsurance administration system; a plurality of support processes, a plurality of hook methods, and a designated order for executing steps in one or more application programs, wherein the steps comprise pre-execution, data entry, data validation, pre-commission, commission, and post-commission;

creating one or more reinsurance business process subclasses from the classes of objects of the reinsurance business process framework, wherein the one or more reinsurance business process subclasses inherit one or more of the hook methods of the reinsurance business process framework;

associating one or more of the support processes with one or more of the reinsurance business process subclasses;

overriding at least one of the hook methods of the reinsurance business process framework to access at least one stage in an execution of one of the reinsurance business processes and to identify a support process to be executed, wherein overriding the at least one hook method comprises overriding a method to be executed during data entry;

combining one or more reinsurance business process subclasses to build one or more application programs for the reinsurance administration system, wherein the order for executing steps in the one or more application programs is the order for the reinsurance business process framework;

creating one or more reinsurance contract objects that represent one or more reinsurance contracts, wherein creating a reinsurance contract object comprises:

identifying one or more inheritable contract objects from the class of objects to represent one or more conditions of a reinsurance contract, wherein the reinsurance contract object is a parent of a section object, and wherein the reinsurance contract comprises the transfer by a first insurer of at least a portion of the risk associated with a primary insurance contract to a second insurer to provide protection to the first insurer against the risk associated with the primary insurance contract;

creating an instance of the inheritable contract object to identify a condition object, wherein the condition object is a child of the section object; and

configuring properties and methods of the condition object consistent with the reinsurance contract;

automatically generating reinsurance process objects as defined by the combined reinsurance business process subclasses when one or more of the application programs are initiated; and

executing at least of the one or more application programs for the reinsurance administration system.

69. The computer readable medium of claim 68, wherein the one or more classes of objects comprises one or more abstract classes, and wherein one or more sub-classes are created from one or more abstract classes.

71. The computer readable medium of claim 68, wherein overriding the at least one hook method comprises replacing the hook method with one or more new methods.

72. The computer readable medium of claim 68, wherein the at least one hook method comprises a method to be executed during initialization of one or more of the application programs.

73. The computer readable medium of claim 68, wherein the at least one hook method that is overridden comprises a method to be executed prior to execution of a data entry support process.

74. The computer readable medium of claim 68, wherein the at least one hook method that is overridden comprises a method to be executed during execution of data entry without a user interface.

75. The computer readable medium of claim 68, wherein the at least one hook method that is overridden comprises a method to be executed prior to execution of a database commit support process.

76. The computer readable medium of claim 68, wherein the at least one hook method that is overridden comprises a method to be executed prior to execution of a database rollback support process.

77. The computer readable medium of claim 68, wherein the at least one hook method that is overridden comprises a method to be executed during validation of data.

79. The computer readable medium of claim 68, wherein the support process to be executed comprises a process for performing precondition checks.

80. The computer readable medium of claim 68, wherein the support process to be executed comprises a process for performing syntax validation.

81. The computer readable medium of claim 68, wherein the support process to be executed comprises a process for a graphical user interface.

82. The computer readable medium of claim 68, wherein the support process to be executed comprises a process for determining the behavior of buttons in a graphical user interface.

83. The computer readable medium of claim 68, wherein the support process to be executed comprises a process for controlling access to information as a function of an access right of a user.

84. The computer readable medium of claim 68, wherein the support process to be executed comprises a process for determining functionality for formatting and displaying windows in a graphical user interface.

85. The computer readable medium of claim 68, wherein the support process to be executed comprises a process for data validation.

86. The computer readable medium of claim 68, wherein the support process to be executed comprises a process for security.

87. The computer readable medium of claim 68, wherein the support process to be executed comprises a process for persistent data storage.

90. The computer readable medium of claim 68, wherein the reinsurance business process framework comprises a process for logging and displaying error messages.

91. The computer readable medium of claim 68, wherein the reinsurance business process framework comprises a process for committing changes to a database.

92. The computer readable medium of claim 68, wherein the program instructions are further computer-executable to implement storing the one or more application programs on a storage device.

95. A method for developing a reinsurance administration system for reinsurance contracts, the method comprising:

obtaining a reinsurance business process framework, wherein the reinsurance business process framework comprises common functionality for one or more reinsurance business processes and wherein the reinsurance business process framework comprises: one or more classes of objects designed for a reinsurance administration system; a plurality of support processes; a plurality of hook methods, and a designated order for executing steps in one or more application programs, wherein the steps comprise pre-execution, data entry, data validation, pre-commission, commission, and post-commission;

creating one or more reinsurance business process subclasses from classes of objects of the reinsurance business process framework, wherein the one or more reinsurance business process subclasses inherit one or more of the hook methods of the reinsurance business process framework;

associating one or more of the support processes with one or more of the reinsurance business process subclasses;

overriding at least one of the hook methods of the reinsurance business process framework to access at least one stage in an execution of one of the reinsurance business processes and to identify a support process to be executed, wherein overriding the at least one hook method comprises overriding a method to be executed during data entry;

combining one or more subclasses to build one or more application programs for the reinsurance administration system, wherein the order for executing steps in the one or more application programs is the order for the reinsurance business process framework;

creating one or more reinsurance contract objects that represent one or more reinsurance contracts, wherein creating a reinsurance contract object comprises:

identifying one or more inheritable contract objects from the class of objects to represent one or more conditions of a reinsurance contract, wherein the reinsurance contract object is a parent of a section object, and wherein the reinsurance contract comprises the transfer by a first insurer of at least a portion of the risk associated with a primary insurance contract to a second insurer to provide protection to the first insurer against the risk associated with the primary insurance contract;

creating an instance of the inheritable contract object to identify a condition object, wherein the condition object is a child of the section object; and

configuring properties and methods of the condition object consistent with the reinsurance contract; and

automatically generating reinsurance process objects as defined by the combined reinsurance business process subclasses when one or more of the application programs are initiated; and

executing at least of the one or more application programs for the reinsurance administration system.

96. The method of claim 95, wherein the one or more classes of objects comprises one or more abstract classes, and wherein one or more sub-classes are created from one or more abstract classes.

98. The method of claim 95, wherein overriding the at least one hook method comprises replacing the hook method with one or more new methods.

99. The method of claim 95, wherein the at least one hook method that is overridden comprises a method to be executed during initialization of one or more of the application programs.

100. The method of claim 95, wherein the at least one hook method that is overridden comprises a method to be executed prior to execution of a data entry support process.

101. The method of claim 95, wherein the at least one hook method that is overridden comprises a method to be executed during execution of data entry without a user interface.

102. The method of claim 95, wherein the at least one hook method that is overridden comprises a method to be executed prior to execution of a database commit support process.

103. The method of claim 95, wherein the at least one hook method that is overridden comprises a method to be executed prior to execution of a database rollback support process.

104. The method of claim 95, wherein the at least one hook method that is overridden comprises a method to be executed during validation of data.

106. The method of claim 95, wherein the reinsurance framework support processes comprise a process for performing precondition checks.

107. The method of claim 95, wherein the one or more support process to be executed comprises a process for performing syntax validation.

108. The method of claim 95, wherein the support process to be executed comprises a process for a graphical user interface.

109. The method of claim 95, wherein the support process to be executed comprises a process for determining the behavior of buttons in a graphical user interface.

110. The method of claim 95, wherein the support process to be executed comprises a process for controlling access to information as a function of an access right of a user.

111. The method of claim 95, wherein the support process to be executed comprises a process for determining functionality for formatting and displaying windows in a graphical user interface.

112. The method of claim 95, wherein the support process to be executed comprise a process for data validation.

113. The method of claim 95, wherein the support process to be executed comprise a process for security.

114. The method of claim 95, wherein the support process to be executed comprises a process for persistent data storage.

117. The method of claim 95, wherein the reinsurance business process framework comprises a process for logging and displaying error messages.

118. The method of claim 95, wherein the reinsurance business process framework comprises a process for committing changes to a database.

119. The method of claim 95, wherein the program instructions are further computer-executable to implement storing the one or more application programs on a storage device.

X. Evidence Appendix

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

XI. Related Proceedings Appendix

No decisions have been rendered in any of the above-identified related proceedings.